Amendment to the Claims

1. (Original) A computational component configured for ~~performing a method, the method comprising~~:

performing at least a first fast Walsh transform on a first set of magnitudes, wherein said set of magnitudes contains a number of magnitudes that is equal to a number of chips in a longest valid symbol;

storing a result of said performing at least a first fast Walsh transform in a first register;

comparing each magnitude comprising said result of performing said at least a first fast Walsh transform to a threshold value; and

replacing each magnitude of said stored result of performing said first fast Walsh transform that is greater than said threshold value with a zero to obtain a first modified result.

2. (Amended) The ~~method~~ computational component of Claim 1, ~~wherein~~ configured for storing said first modified result ~~is stored~~ in said first register.

3. (Amended) The ~~method~~ computational component of Claim 1, further ~~comprising~~ configured for:

performing at least a second fast Walsh transform on said first modified result;

storing a result of performing said at least a second fast Walsh transform in said first register;

comparing each magnitude comprising said result of performing said second fast Walsh transform to said threshold value; and

replacing each magnitude of said stored result of performing said at least a second fast Walsh transform that is greater than said threshold value with a zero to obtain a second modified result.

4. (Amended) The ~~method~~ computational component of Claim 3, ~~wherein~~ configured for storing said second modified result ~~is stored~~ in said first register.

5. (Amended) The ~~method~~ computational component of Claim 1, further ~~comprising~~ configured for:

performing an $(n-1)^{th}$ fast Walsh transform on a previously calculated modified result;

storing a result of performing said $(n-1)^{th}$ fast Walsh transform in a register;

comparing each magnitude comprising said result of performing said $(n-1)^{th}$ fast Walsh transform to said threshold value;

replacing each magnitude of said stored result of performing said $(n-1)^{th}$ fast Walsh transform that is greater than said threshold value with a zero to obtain an $(n-1)^{th}$ modified result;

performing an n^{th} fast Walsh transform on said $(n-1)^{th}$ modified result;

storing a result of performing said n^{th} fast Walsh transform in a register;

comparing each magnitude comprising said result of performing said n^{th} fast Walsh transform to said threshold value; and

replacing each magnitude of said stored result of performing said n^{th} fast Walsh transform that is greater than said threshold value with a zero to obtain an n^{th} modified result.

6. (Amended) The ~~method~~ computational component of Claim 5, ~~wherein~~ configured for storing said results of performing said $(n-1)^{th}$ and said n^{th} fast Walsh transforms ~~are stored~~ in said first register.

7. (Amended) The ~~method~~ computational component of Claim 6, ~~wherein~~ configured for not storing said result of performing said $(n-1)^{th}$ fast Walsh transform ~~are not stored~~ in said first register simultaneously with said result of performing said n^{th} fast Walsh transform.

8. (Amended) The ~~method~~ computational component of Claim 5, wherein said register in which each of said $(n-1)^{th}$ modified result and said n^{th} modified result is stored in comprises said first register.

9. (Amended) The ~~method~~ computational component of Claim 5, wherein said previously calculated modified result comprises said first modified result.

10. (Amended) The ~~method~~ computational component of Claim 1, further ~~comprising~~ configured for:

storing each magnitude comprising said result of performing said n^{th} fast Walsh transform having a magnitude that is not less than said threshold value in a second register; and

storing a zero for magnitudes comprising said result of performing said n^{th} fast Walsh transform having a magnitude that is less than said threshold value in said second register, wherein said second register comprises a number of magnitudes that is equal to said number of chips in a longest valid symbol.

11. (Amended) The ~~method~~ computational component of Claim 10, wherein said storing a zero comprises replacing magnitudes stored in said second register having a magnitude that is not greater than said threshold value with a zero.

12. (Amended) The ~~method~~ computational component of Claim 10, wherein said n^{th} fast Walsh transform corresponds to a Walsh code set for symbols of a valid length.

13. (Amended) The ~~method~~ computational component of Claim 10, further ~~comprising~~ configured for:

storing said magnitude comprising said result of performing said $(n-1)^{th}$ fast Walsh transform having a magnitude that is greater than said threshold value in a third register; and

storing a zero for magnitudes comprising said result of performing said $(n-1)^{th}$ fast Walsh transform having a magnitude that is not greater than said threshold value in said third register, wherein said third register comprises a number of magnitudes that is equal to said number of chips in a longest valid symbol.

14. (Amended) The ~~method~~ computational component of Claim 13, wherein said $(n-1)^{th}$ fast Walsh transform corresponds to a Walsh code set for symbols of at least a minimum valid length.

15. (Amended) The ~~method~~ computational component of Claim 10, wherein said second register comprises a number of values equal to said number of chips in a longest valid symbol.

16. (Amended) The ~~method~~ computational component of Claim 13, further ~~comprising~~ configured for:

adding said value in said second register to a product equal to said value in said third register multiplied by 2 to obtain a composite interference vector.

17. (Amended) The ~~method~~ computational component of Claim 16, wherein said n^{th} fast Walsh transform corresponds to a Walsh code set for symbols of a maximum valid length.

18. (Amended) The ~~method~~ computational component of Claim 16, further ~~comprising~~ configured for:

    applying said composite interference vector to a received signal stream to create an interference canceled signal stream.

19. (Amended) The ~~method~~ computational component of Claim 1, wherein said threshold value is derived from a magnitude of a selected received channel within a signal stream from which said first set of magnitudes are obtained.

20. (Amended) The ~~method~~ computational component of Claim 1, wherein said threshold value comprises a value derived from a magnitude of a sync channel.

21. (Amended) The ~~method~~ computational component of Claim 1, wherein said threshold value is equal to said magnitude of a sync channel.

22. (Amended) The ~~method~~ computational component of Claim 1, wherein said threshold value is a preselected value.

23. (Amended) The ~~method~~ computational component of Claim 1, wherein n is a number of fast Walsh transforms performed and is equal to log N, where N is the number of valid traffic channels.

24. (Amended) The ~~method~~ computational component of Claim 1, ~~wherein said computational component comprises~~ comprising a computer readable storage medium containing instructions ~~for performing the method~~.

25. (Amended) The ~~method~~ computational component of Claim 1, ~~wherein said computational component comprises~~ comprising a logic circuit.

26. (Original) A method for calculating interference calculation values, comprising:

    receiving a signal stream comprising a plurality of channels;

    despreading said signal stream by applying a despreading code;

    obtaining a first number of chip values from said despread signal stream, wherein said first number is equal to a number of chips included in a longest valid symbol;

performing a fast Walsh transform on said first number of chip values to obtain a first set of transformed values, wherein said first result includes a first number of elements equal to said first number of chip values;

comparing a value of each of said first number of elements of said first set of transformed values to a threshold; and

creating a first modified set of values, wherein for each element of said first set of transformed values:

in response to a first result of said comparison, a value of said element is changed to a zero;

in response to a second result of said comparison, a value of said element is not changed to a zero.

27. (Original) The method of Claim 26, wherein said not replacing a value of said element with a zero comprises leaving said value of said element intact.

28. (Original) The method of Claim 26, further comprising:

performing a fast Walsh transform on a previously created modified set of values to obtain an $(n-1)^{th}$ set of transformed values, wherein said $(n-1)^{th}$ set of transformed values includes said first number of elements;

comparing a value of each of said first number of elements of said $(n-1)^{th}$ set of transformed values to a threshold;

creating an $(n-1)^{th}$ modified set of values, wherein for each element of said $(n-1)^{th}$ set of transformed values:

in response to a first result of said comparison, a value of said element is changed to a zero; and

in response to a second result of said comparison, a value of said element is not changed to a zero.

29. (Original) The method of Claim 28, wherein said previously created modified set of values comprises said first result.

30. (Original) The method of Claim 28, further comprising:

performing a fast Walsh transform on said $(n-1)^{th}$ modified set of values to obtain an n^{th} set of transformed values, wherein said n^{th} set of transformed values includes said first number of elements;

comparing a value of each of said first number of elements of said n^{th} set of transformed values to a threshold;

creating an n^{th} modified set of values, wherein for each element of said n^{th} set of transformed values:

in response to a first result of said comparison, a value of said element is changed to a zero; and

in response to a second result of said comparison, a value of said element is not changed to a zero.

31. (Original) The method of Claim 30, further comprising:

creating a first composite interference vector component, wherein a value of each element of said n^{th} modified set of values is compared to a threshold, and wherein for each element of said n^{th} modified set of values:

in response to a first result of said comparison, not changing a value of said element to a zero, and

in response to a second result of said comparison, changing a value of said element to a zero.

32. (Original) The method of Claim 31, further comprising:

creating a second composite interference vector component, wherein a value of each element of said $(n-1)^{th}$ modified set of values is compared to a threshold, and wherein for each element of said $(n-1)^{th}$ modified set of values:

in response to a first result of said comparison, not changing a value of said element to a zero, and

in response to a second result of said comparison, changing a value of said element to a zero.

33. (Original) The method of Claim 32, further comprising:

combining said first and second composite interference vector components to one another to create a composite interference vector.

34. (Original) The method of Claim 32, further comprising:

scaling said second composite interference vector component to obtain a scaled second composite interference vector component; and

adding said first composite interference vector component to said scaled second composite interference vector component to obtain a composite interference vector.

35. (Original) The method of Claim 34, further comprising:

projecting said composite interference vector onto a received signal stream to obtain an interference cancelled signal.

36. (Original) The method of Claim 26, wherein said changing a value of an element comprises replacing said value in a register.

37. (Original) An apparatus for determining communication channel values, comprising:

means for receiving a signal path;

means for performing at least a first fast Walsh transform on a selected set of element amplitudes that is one of received as part of said signal path or received as part of said signal path and modified, wherein a first set of modified element amplitudes is obtained;

means for comparing said first set of modified element amplitudes to a threshold; and

first means for storing a channel estimate, wherein said channel estimate includes an element amplitude for an element having an amplitude that does not exceed said threshold and a zero for an element having an amplitude that exceeds said threshold.

38. (Original) The apparatus of Claim 37, wherein said selected set of element amplitudes is modified by providing said selected set of elements to means for performing a fast Walsh transform prior to providing said resulting modified element amplitudes to said means for performing at least a first fast Walsh transform.

39. (Original) The apparatus of Claim 37, further comprising:

means for performing at least a first fast Walsh transform on said channel estimate, wherein a second set of modified element amplitudes is obtained.

40. (Original) The apparatus of Claim 39, further comprising:

means for comparing said second set of modified element amplitudes to a threshold.

41. (Original) The apparatus of Claim 37, further comprising:

means for storing an interference vector precursor, wherein said interference vector precursor includes an element amplitude for an element having an amplitude that

exceeds said threshold and a zero for an element having an amplitude that does not exceed said threshold.

42. (Original) The apparatus of Claim 41, further comprising:

    means for performing at least a first fast Walsh transform on said interference vector precursor to obtain an interference vector.

43. (Original) The apparatus of Claim 42, further comprising:

    means for storing said interference vector.

44. (Original) The apparatus of Claim 42, further comprising:

    means for scaling an interference vector.

45. (Original) The apparatus of Claim 44, further comprising means for combining a plurality of interference vectors to form a composite interference vector.

46. (Original) A receiver device, comprising:

    a fast Walsh transform module operable to perform a selected fast Walsh transform stage on a set of values;

    a comparator operable to compare each value output from said fast Walsh transform module to a threshold;

    a first memory register operable to store element values output from said comparator as having a value less than said threshold; and

    a second memory register operable to store element values output from said comparator as having a value not less than said threshold.

47. (Original) The device of Claim 46, wherein said comparator is additionally operable to output a zero for storing in said first memory register in place of element values having a value greater than said threshold.

48. (Original) The device of Claim 46, wherein said comparator is additionally operable to output a zero for storing in said second memory register in place of element values having a value less than said threshold.

49. (Original) The device of Claim 46, further comprising:

    a multiplexer operable to provide said element values stored in said second memory to said fast Walsh transform module, said fast Walsh transform module additionally being operable to perform at least a first fast Walsh transform on said stored element values to obtain an interference vector.

50. (Original) The device of Claim 49, further comprising:

scalar operable to multiply said interference vector by a selected value.

51. (Original) The device of Claim 50, further comprising a summer operable to add a plurality of scaled interference vectors to obtain a composite interference vector.

52. (New) A computer readable storage medium containing instructions configured for:

performing at least a first fast Walsh transform on a first set of magnitudes, wherein said set of magnitudes contains a number of magnitudes that is equal to a number of chips in a longest valid symbol;

storing a result of said performing at least a first fast Walsh transform in a first register;

comparing each magnitude comprising said result of performing said at least a first fast Walsh transform to a threshold value; and

replacing each magnitude of said stored result of performing said first fast Walsh transform that is greater than said threshold value with a zero to obtain a first modified result.

53. (New) The computer readable storage medium of Claim 52, wherein the instructions are configured for storing said first modified result in said first register.

54. (New) The computer readable storage medium of Claim 52, wherein the instructions are further configured for:

performing at least a second fast Walsh transform on said first modified result;

storing a result of performing said at least a second fast Walsh transform in said first register;

comparing each magnitude comprising said result of performing said second fast Walsh transform to said threshold value; and

replacing each magnitude of said stored result of performing said at least a second fast Walsh transform that is greater than said threshold value with a zero to obtain a second modified result.

55. (New) The computer readable storage medium of Claim 54, wherein the instructions are configured for storing said second modified result in said first register.

56. (New) The computer readable storage medium of Claim 52, wherein the instructions are further configured for:

performing an $(n-1)^{th}$ fast Walsh transform on a previously calculated modified result;

storing a result of performing said $(n-1)^{th}$ fast Walsh transform in a register;

comparing each magnitude comprising said result of performing said $(n-1)^{th}$ fast Walsh transform to said threshold value;

replacing each magnitude of said stored result of performing said $(n-1)^{th}$ fast Walsh transform that is greater than said threshold value with a zero to obtain an $(n-1)^{th}$ modified result;

performing an n^{th} fast Walsh transform on said $(n-1)^{th}$ modified result;

storing a result of performing said n^{th} fast Walsh transform in a register;

comparing each magnitude comprising said result of performing said n^{th} fast Walsh transform to said threshold value; and

replacing each magnitude of said stored result of performing said n^{th} fast Walsh transform that is greater than said threshold value with a zero to obtain an n^{th} modified result.

57. (New) The computer readable storage medium of Claim 56, wherein the instructions are configured for storing said results of performing said $(n-1)^{th}$ and said n^{th} fast Walsh transforms in said first register.

58. (New) The computer readable storage medium of Claim 57, wherein the instructions are configured for not storing said result of performing said $(n-1)^{th}$ fast Walsh transform in said first register simultaneously with said result of performing said n^{th} fast Walsh transform.

59. (New) The computer readable storage medium of Claim 56, wherein said register in which each of said $(n-1)^{th}$ modified result and said n^{th} modified result is stored in comprises said first register.

60. (New) The computer readable storage medium of Claim 56, wherein said previously calculated modified result comprises said first modified result.

61. (New) The computer readable storage medium of Claim 52, wherein the instructions are further configured for:

storing each magnitude comprising said result of performing said n^{th} fast Walsh transform having a magnitude that is not less than said threshold value in a second register; and

storing a zero for magnitudes comprising said result of performing said n^{th} fast Walsh transform having a magnitude that is less than said threshold value in said second register, wherein said second register comprises a number of magnitudes that is equal to said number of chips in a longest valid symbol.

62. (New) The computer readable storage medium of Claim 61, wherein said storing a zero comprises replacing magnitudes stored in said second register having a magnitude that is not greater than said threshold value with a zero.

63. (New) The computer readable storage medium of Claim 61, wherein said n^{th} fast Walsh transform corresponds to a Walsh code set for symbols of a valid length.

64. (New) The computer readable storage medium of Claim 61, wherein the instructions are further configured for:

storing said magnitude comprising said result of performing said $(n-1)^{th}$ fast Walsh transform having a magnitude that is greater than said threshold value in a third register; and

storing a zero for magnitudes comprising said result of performing said $(n-1)^{th}$ fast Walsh transform having a magnitude that is not greater than said threshold value in said third register, wherein said third register comprises a number of magnitudes that is equal to said number of chips in a longest valid symbol.

65. (New) The computer readable storage medium of Claim 64, wherein said $(n-1)^{th}$ fast Walsh transform corresponds to a Walsh code set for symbols of at least a minimum valid length.

66. (New) The computer readable storage medium of Claim 61, wherein said second register comprises a number of values equal to said number of chips in a longest valid symbol.

67. (New) The computer readable storage medium of Claim 64, wherein the instructions are further configured for:

adding said value in said second register to a product equal to said value in said third register multiplied by 2 to obtain a composite interference vector.

68. (New) The computer readable storage medium of Claim 67, wherein said n^{th} fast Walsh transform corresponds to a Walsh code set for symbols of a maximum valid length.

69. (New) The computer readable storage medium of Claim 67, wherein the instructions are further configured for:

    applying said composite interference vector to a received signal stream to create an interference canceled signal stream.

70. (New) The computer readable storage medium of Claim 52, wherein said threshold value is derived from a magnitude of a selected received channel within a signal stream from which said first set of magnitudes are obtained.

71. (New) The computer readable storage medium of Claim 52, wherein said threshold value comprises a value derived from a magnitude of a sync channel.

72. (New) The computer readable storage medium of Claim 52, wherein said threshold value is equal to said magnitude of a sync channel.

73. (New) The computer readable storage medium of Claim 52, wherein said threshold value is a preselected value.

74. (New) The computer readable storage medium of Claim 52, wherein n is a number of fast Walsh transforms performed and is equal to log N, where N is the number of valid traffic channels.

75. (New) The computer readable storage medium of Claim 52, comprising a logic circuit.

76. (New) A logic circuit configured for:

    performing at least a first fast Walsh transform on a first set of magnitudes, wherein said set of magnitudes contains a number of magnitudes that is equal to a number of chips in a longest valid symbol;

    storing a result of said performing at least a first fast Walsh transform in a first register;

    comparing each magnitude comprising said result of performing said at least a first fast Walsh transform to a threshold value; and

    replacing each magnitude of said stored result of performing said first fast Walsh transform that is greater than said threshold value with a zero to obtain a first modified result.

77. (New) The logic circuit of Claim 76, configured for storing said first modified result in said first register.

78. (New) The logic circuit of Claim 76, further configured for:

performing at least a second fast Walsh transform on said first modified result;

storing a result of performing said at least a second fast Walsh transform in said first register;

comparing each magnitude comprising said result of performing said second fast Walsh transform to said threshold value; and

replacing each magnitude of said stored result of performing said at least a second fast Walsh transform that is greater than said threshold value with a zero to obtain a second modified result.

79. (New) The logic circuit of Claim 78, configured for storing said second modified result in said first register.

80. (New) The logic circuit of Claim 76, further configured for:

performing an $(n-1)^{th}$ fast Walsh transform on a previously calculated modified result;

storing a result of performing said $(n-1)^{th}$ fast Walsh transform in a register;

comparing each magnitude comprising said result of performing said $(n-1)^{th}$ fast Walsh transform to said threshold value;

replacing each magnitude of said stored result of performing said $(n-1)^{th}$ fast Walsh transform that is greater than said threshold value with a zero to obtain an $(n-1)^{th}$ modified result;

performing an n^{th} fast Walsh transform on said $(n-1)^{th}$ modified result;

storing a result of performing said n^{th} fast Walsh transform in a register;

comparing each magnitude comprising said result of performing said n^{th} fast Walsh transform to said threshold value; and

replacing each magnitude of said stored result of performing said n^{th} fast Walsh transform that is greater than said threshold value with a zero to obtain an n^{th} modified result.

81. (New) The logic circuit of Claim 80, configured for storing said results of performing said $(n-1)^{th}$ and said n^{th} fast Walsh transforms in said first register.

82. (New) The logic circuit of Claim 81, configured for not storing said result of performing said $(n-1)^{th}$ fast Walsh transform in said first register simultaneously with said result of performing said n^{th} fast Walsh transform.

83. (New) The logic circuit of Claim 80, wherein said register in which each of said $(n-1)^{th}$ modified result and said n^{th} modified result is stored in comprises said first register.

84. (New) The logic circuit of Claim 80, wherein said previously calculated modified result comprises said first modified result.

85. (New) The logic circuit of Claim 76, further configured for:

storing each magnitude comprising said result of performing said n^{th} fast Walsh transform having a magnitude that is not less than said threshold value in a second register; and

storing a zero for magnitudes comprising said result of performing said n^{th} fast Walsh transform having a magnitude that is less than said threshold value in said second register, wherein said second register comprises a number of magnitudes that is equal to said number of chips in a longest valid symbol.

86. (New) The logic circuit of Claim 85, wherein said storing a zero comprises replacing magnitudes stored in said second register having a magnitude that is not greater than said threshold value with a zero.

87. (New) The logic circuit of Claim 85, wherein said n^{th} fast Walsh transform corresponds to a Walsh code set for symbols of a valid length.

88. (New) The logic circuit of Claim 85, further configured for:

storing said magnitude comprising said result of performing said $(n-1)^{th}$ fast Walsh transform having a magnitude that is greater than said threshold value in a third register; and

storing a zero for magnitudes comprising said result of performing said $(n-1)^{th}$ fast Walsh transform having a magnitude that is not greater than said threshold value in said third register, wherein said third register comprises a number of magnitudes that is equal to said number of chips in a longest valid symbol.

89. (New) The logic circuit of Claim 88, wherein said $(n-1)^{th}$ fast Walsh transform corresponds to a Walsh code set for symbols of at least a minimum valid length.

90. (New) The logic circuit of Claim 85, wherein said second register comprises a number of values equal to said number of chips in a longest valid symbol.

91. (New) The logic circuit of Claim 88, further configured for:

    adding said value in said second register to a product equal to said value in said third register multiplied by 2 to obtain a composite interference vector.

92. (New) The logic circuit of Claim 91, wherein said n^{th} fast Walsh transform corresponds to a Walsh code set for symbols of a maximum valid length.

93. (New) The logic circuit of Claim 91, further configured for:

    applying said composite interference vector to a received signal stream to create an interference canceled signal stream.

94. (New) The logic circuit of Claim 76, wherein said threshold value is derived from a magnitude of a selected received channel within a signal stream from which said first set of magnitudes are obtained.

95. (New) The logic circuit of Claim 76, wherein said threshold value comprises a value derived from a magnitude of a sync channel.

96. (New) The logic circuit of Claim 76, wherein said threshold value is equal to said magnitude of a sync channel.

97. (New) The logic circuit of Claim 76, wherein said threshold value is a preselected value.

98. (New) The logic circuit of Claim 76, wherein n is a number of fast Walsh transforms performed and is equal to log N, where N is the number of valid traffic channels.